running modelE with -O0 -g traceback ...

wputman 22 posts since

Aug 16, 2007 I'm wondering if anyone has run modelE on discover with -O0 instead of -O2. And if so can you run with ifort debugging options (-g traceback -check uninit -check bounds)? My initial attempts have failed, I get a number of exits generated by the model saying land/lake fields have values of zero. Tags: modele, discover, optimization, ifort

ialeinov *12 posts since*
Sep 20, 2007 1. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 1, 2008 8:27 PM

Sorry for late reply on this. Apparently this was also noticed by other people but they didn't complain loud enough. I tried to investigate this and narrowed the problem to a single function CreateDist_MPI_Type() inside DOMAIN_DECOMP.f . If this function (when moved into a separate file) is compiled with -O2 everything works correctly. If it is compiled with -O0 the model crashes. The cause of the crash is the garbage (usually zeros) which is returned by MPI_ScatterV (which uses MPI types created by CreateDist_MPI_Type()). I don't see anything wrong with this function but I can't claim that I understand this stuff (MPI structures). Could be compiler bug... But, really, this is the third time this function is causing us trouble (the first two being memory leaks in both SCALI and OPENMPI). Would it be too much work to rewrite the code so that MPI_ScatterV doesn't need to create new MPI types?

clune *113 posts since*
May 31, 2007 2. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 1, 2008 10:56 PM

in response to: ialeinov Interesting - these are far more specific symptoms, and I'd like to track them down. (BTW, if you use the MPP layer, you should be skipping the scatter entirely, if I understand correctly.)

I tried building E1M20 with ESMF this evening, but I get the following run time error:

Vegetation data is missing at some cells <<<

Looking at my .PRT file, the problem seems pretty pervasive. Does something else need to be updated?

ialeinov *12 posts since*
Sep 20, 2007 3. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 1, 2008 11:33 PM

Vegetation data is missing at some cells <<< That's what I'm getting if I compile the model with -O0 . That's because some scattered arrays are filled with 0's instead of real data and that fails the sanity check during the initialization of land surface model. Everything seems to work fine when compiled with -O2 .

ialeinov *12 posts since*
Sep 20, 2007 4. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 2, 2008 12:11 AM

Well, I guess there was no mistery here. ext_lb, base_byte_len have to be "MPI_Aint". Will commit fix in a moment....

clune *113 posts since*
May 31, 2007 5. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 2, 2008 10:11 AM

in response to: ialeinov Hmm. Good catch. I would have thought that I would have checked these data types more thoroughly when I first implemented this section. Apparently I was getting lucky that the earlier releases were using default integers, or some other similar coincidence. I don't see your commits in the repository yet, so forgive me if I manage to test/commit this first. (Checking later - I now see that you already committed.)

I guess this could also explain the memory leaks, though it is not immediately obvious just how that would be manifested.

ialeinov *12 posts since*
Sep 20, 2007 6. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 2, 2008 5:05 PM

Actually this was relatively recent bug (on Aug 19 calls to "old" functions were replaced by those to "new" functions which require integer*8). It had nothing to do with SCALI memory leak and I doubt that it is relates to OPENMPI memory leak. One has to pass two addresses to two 8-byte objects and instead one passes addresses to 4-byte objects. This overwrites memory (4 bytes) right after that object (most probably some other local variable). On little-endian machine it will be overwritten with 0's (while 4-byte object will still get a correct value). I don't see how this can create a memory leak. You either break the program completely or it continues to work correctly (if you overwrite a variable which is not needed).

All this mess, of course, could not happen if the program was written in C, where all interfaces are well defined and checked automatically. Sure, core MPI developers didn't even think that these functions could be called from Fortran. That's why one needs MPI_ADDRESS_KIND hacks...

clune *113 posts since*
May 31, 2007 7. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 2, 2008 5:28 PM

in response to: ialeinov Thanks - my feelings of guilt are significantly reduced.

Of course, in theory there is the "USE MPI" mechanism to check these interfaces, but in practice not enough implementations provide that module to be truly portable. We might want to re-evaluate that - perhaps all the relevant MPI vendors now have a robust MPI mod for the interfaces.

ialeinov *12 posts since*
Sep 20, 2007 8. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 2, 2008 6:02 PM

I don't see how it is practically possible. Structure of .mod files is undocumented and is different not only for different vendors but also for different versions of compiler from the same vendor. I guess what one could do is to create include files with ITERFACE blocks for all MPI calls (similar to header files with function prototypes in C). But for some reason this approach is not used...

clune *113 posts since*
May 31, 2007 9. **Re: running modelE with -O0 -g traceback -check uninit -check bounds** Dec 2, 2008 7:14 PM

in response to: ialeinov This does not need to be done directly by providing a .mod file. Typically the MPI vendor provides a module that is compiled with whichever compiler you are trying to integrate with. Something similar is already happening with the Fortran wrappers to the underlying C interfaces. The discover admins are rebuilding MPI for each compiler (and major revision). The system has some kinks no doubt, but it certainly should have caught this integer KIND problem immediately.